

## Chapter 10

# Towards Understanding the Origin of Genetic Languages

Apoorva D. Patel

*“...four and twenty blackbirds baked in a pie ...”*

Molecular biology is a nanotechnology that works—it has worked for billions of years and in an amazing variety of circumstances. At its core is a system for acquiring, processing and communicating information that is universal, from viruses and bacteria to human beings. Advances in genetics and experience in designing computers have taken us to a stage where we can understand the optimisation principles at the root of this system, from the availability of basic building blocks to the execution of tasks. The languages of DNA and proteins are argued to be the optimal solutions to the information processing tasks they carry out. The analysis also suggests simpler predecessors to these languages, and provides fascinating clues about their origin. Obviously, a comprehensive unraveling of the puzzle of life would have a lot to say about what we may design or convert ourselves into.

### 10.1 The Meaning of It All

I am going to write about some of the defining characteristics of life. Philosophical issues always arise in discussions regarding life, and I cannot avoid that. But let me state at the outset that such issues are not the purpose of my presentation. I am going to look at life as an exercise in information theory, and extend the analysis as far as possible.

Let me begin with the textbook answer to the question made famous

by Schrödinger [Schrödinger (1944)]: *What is life?* Life is fundamentally a non-equilibrium process, commonly characterised in terms of two basic phenomena. One is “metabolism”. Many biochemical processes are needed to sustain a living organism. Running these processes requires a continuous supply of free energy, which is extracted from the environment. (Typically this energy is in electromagnetic or chemical form, but its ultimate source is gravity—the only interaction in the universe that is not in equilibrium.) The other is “reproduction”. A particular physical structure cannot survive forever, because of continuous environmental disturbances and consequent damages. So life perpetuates itself by a succession of generations.

It is obvious that both these phenomena are sustaining and protecting and improving something, often against the odds. So let us figure out what is it that is being sustained and protected and improved.

All living organisms are made up of atoms. These atoms are fantastically indestructible. In all the biochemical processes, they just get rearranged in different ways. Each of us would have a billion atoms that once belonged to the Buddha, or Genghis Khan, or Isaac Newton—a sobering or exciting realisation depending on one’s frame of mind! We easily see that it is not the atoms themselves but their arrangements in complex molecules, which carry biochemical information. In the flow of biochemical processes, living organisms synthesise and break up various molecules, by altering atomic arrangements. The biochemical information resides in what molecules to use where, when and how. Characterisation of this information is rather abstract, but central to the understanding of life. To put it succinctly:

Hardware is recycled, while software is refined!

At the physical level, atoms are shuffled, molecules keep on changing, and life goes on. At the abstract level, it is the manipulation and preservation of information that requires construction of complex structures. Information is not merely “a” property of life—it is “the” basis of life.

Now information is routinely quantified as entropy of the possible forms a message could take [Shannon (1948)]. What the living organisms require, however, is not mere information but information with meaning. A random arrangement of components (e.g. a gas) can have large information, but it is not at all clear how that can be put to any use. The molecules of life are destined to carrying out specific functions, and they have to last long enough to execute their tasks. The meaning of biological information is carried by the chemical properties of the molecules, and a reasonably stable

cellular environment helps in controlling the chemical reactions. What the living organisms use is “knowledge”,

$$\text{Knowledge} = \text{Information} + \text{Interpretation}.$$

Knowledge has to be communicated using a language. A language uses a set of building blocks (e.g. letters of an alphabet) whose meaning is fixed, and whose variety of arrangements (invariably aperiodic) compose different messages. It is the combination of information and interpretation that makes languages useful in practice.

Thus to understand how living organisms function, we need to focus on the corresponding languages whose interpretation remains fixed, while all manipulations of information processing go on. A practical language is never constructed arbitrarily—criteria of efficiency are always involved. These criteria are necessarily linked to the tasks to be implemented using the language, and fall into two broad categories. One is the stability of the meaning, i.e. protection against error causing fluctuations. And the other is the efficient use of physical resources, i.e. avoidance of unnecessary waste of space, time, energy etc. while conveying a message. The two often impose conflicting demands on the language, and the question to investigate is: *Is there an optimal language for a given task, and if so how can we find it?* From the point of view of a computer designer, the question has two parts:

Software: What are the tasks? What are the algorithms?

Hardware: How are the operations physically implemented?

It goes without saying that the efficiency of a language depends both on the software and the hardware.

In the computational complexity analysis, space and time resources are often traded off against each other, and algorithms are categorized as polynomial or non-polynomial (usually exponential). In the biological context, however, the efficiency considerations are not quite the same. Time is highly precious, while space is fairly expendable. Biological systems can sense small differences in population growth rates, and even an advantage of a fraction of a percent is sufficient for one species to overwhelm another over many generations. Spatial resources are frequently wasted, that too on purpose. For instance, how many seeds does a plant produce, when just a single one can ensure continuity of its lineage? It must not be missed that this wastefulness leads to competition and Darwinian selection.

Before going on to the details of the genetic languages, here is a quick summary of the components making up the biochemical machinery of liv-

ing organisms, at different scales. A framework for understanding genetic languages must incorporate this hierarchical structure.

Atoms	H,C,N,O, and infrequently P,S
Nucleotide bases and amino acids	10-20 atoms
Peptides and drugs	40-100 atoms
Proteins	100-1000 amino acids
Genomes	$10^3\text{-}10^9$ nucleotide base pairs
Size	1 nm (molecules)- $10^4$ nm (cells)

Gene and protein databases have been accumulating a lot of data, which can be used to test hypotheses and consequences of specific choice of languages.

To summarise, the aim of this article is to understand the physical and the evolutionary reasons for (a) the specific genetic languages, and (b) their specific realisations. A tiny footnote is that such an understanding would have a bearing on the probability of finding life elsewhere in the universe and then characterising it.

## 10.2 Lessons of Evolution

Evolution is the centrepiece of biology. It has been the cause of many controversies, mainly because it is almost imperceptible—the evolutionary timescales are orders of magnitude larger than the lifetimes of individual living organisms. But it is the only scientific principle that provides a unifying framework encompassing all forms of life, from the simple origin to an amazing variety. We need to understand the forces governing the direction of evolution, in order to comprehend where we came from as well as what the future may have in store for us.

Genetic information forms the quantitative underpinning of evolution. Certain biological facts regarding genetic languages are well-established:

- (1) Languages of genes and proteins are universal.  
The same 4 nucleotide bases and 20 amino acids are used in DNA, RNA and proteins, all the way from viruses and bacteria to human beings. This is despite the fact that other nucleotide bases and amino acids exist in living cells. This clearly implies that selection of specific languages has taken place.
- (2) Genetic information is encoded close to data compression limit and maximal packing.  
This indicates that optimisation of information storage has taken place.

- (3) Evolution occurs through random mutations, which are local changes in the genetic sequence. In the long run, however, only a small fraction of the mutations survive—those proving advantageous to the organisms. This optimising mechanism is labeled Darwinian selection, i.e. competition for limited resources leading to survival of the fittest.

Over the years, many attempts have been made to construct evolutionary scenarios that can explain the universality of genetic languages. They can be broadly classified into two categories. One category is the “frozen accident” hypothesis [Crick (1968)], i.e. the language somehow came into existence, and became such a vital part of life’s machinery that any change in it would be highly deleterious to living organisms. This requires the birth of the genetic machinery to be an extremely rare event, without sufficient time to explore other possibilities. There is not much room for analysis in this ready-made solution. I do not subscribe to it, and instead argue for the other category. That is the “optimal solution” end-point [Patel (2003)], i.e. the language arrived at its best form by trial and error, and it did not change thereafter, because any change in it would make the information processing less competitive. This requires the evolution of genetic machinery to have sufficient scope to generate many possibilities, and subsequent competition amongst them whence the optimal solution wins over the rest.

It should be noted that the existence of an optimising mechanism does not make a choice between the two categories clear-cut. The reason is that a multi-parameter optimisation manifold generically has a large number of minima and maxima, and an optimisation process relying on only local changes often gets trapped in local minima of the undulating manifold without reaching the global optimum. In such situations, the initial conditions and history of evolution become crucial in deciding the outcome of the process, and typically there arise several isolated surviving candidates. The globally optimal solution is certainly easier to reach, when the number of local minima is small and/or the range of exploratory changes is large. The extent of optimisation is therefore critically controlled by the ratio of time available for exploration of various possibilities to the transition time amongst them. For the genetic machinery to have reached its optimal form, the variety of possibilities thrown up by the primordial soup must have had a simple and quick winner.

The procedure of optimisation needs a process of change, and a process of selection. The former is intrinsic, the latter is extrinsic, and the two take place at different levels in biology. Indeed the difference between the two

provides much ammunition for debates involving choice vs. environment, or nature vs. nurture. The changes are provided by mutations, which occur essentially randomly at the genetic level. That describes the genotype. The selection takes place by the environmental pressure at the level of whole organisms. It is not at all random, rather it is biased towards short-term survival (till reproduction). That describes the phenotype. We have good reasons to believe that the primitive living organisms were unicellular, without a nucleus, with small genomes, and having a simple cellular machinery. In such systems, the genotype and phenotype levels are quite close, and the early evolution can easily be considered a direct optimisation problem.

Before exploring what could have happened in the early stages of evolution, let us also briefly look at the direction in which it has continued. The following table summarises how the primitive unicellular organisms progressed to the level of humans (certainly the most developed form of life in our own point of view), using different physical resources to process information at different levels.

Organism	Messages	Physical Means
Single cell	Molecular (DNA, Proteins)	Chemical bonds, Diffusion
Multicellular	Electrochemical (Nervous system)	Convection, Conduction
Families, Societies	Imitation, Teaching, Languages	Light, Sound
Humans	Books, Computers, Telecommunication	Storage devices, Electromagnetic waves
Gizmos or Cyborgs ?	Databases	Merger of brain and computer

It is clear that evolution has progressively discovered higher levels of communication mechanisms, whereby the communication range has expanded (both in space and time), the physical contact has reduced, abstraction has increased, succinct language forms have arisen and complex translation machinery has been developed. More interesting is the manner in which all this has been achieved, with cooperation (often with division of labour) gradually replacing competition. This does not contradict Darwinian selection—it is just that the phenotype level has moved up, and components of a phenotype are far more likely to cooperate than compete. The mathematical formulation underlying this behaviour is “repeated games”, with no

foresight but with certain amount of memory [Aumann (2006)].

The evolutionary features useful for the purpose of this article are:

- The older and lower information processing levels are far better optimised than the more recent higher levels. This is a consequence of the fact that in the optimisation process the lower levels had less options to deal with and more time to settle on a solution.
- The capacity of gathering, using and communicating knowledge has grown by orders of magnitude in the course of evolution. Indeed one can surmise that, in the long run, the reach of knowledge overwhelms physical features in deciding survival fitness.

*Knowledge is the essential driving force behind evolution,  
providing a clear direction even when the goal remains unclear.*

### 10.3 Genetic Languages

Let us now return to analysing the lowest level of information processing, i.e. the genetic languages. There are two of them—the language of DNA and RNA with an alphabet of four nucleotide bases, and the language of proteins with an alphabet of twenty amino acids. The tasks carried out by both of them are quite specific and easy to identify.

(1) The essential job of DNA and RNA is to sequentially assemble a chain of building blocks on top of a pre-existing master template. One can call DNA the read-only-memory of living organisms. When not involved in the replication process, the information in DNA remains idle in a secluded and protected state.

(2) Proteins are structurally stable molecules of various shapes and sizes, with precise locations of active chemical groups. They carry out various functions of life by highly selective binding to other molecules. Molecular interactions are weak and extremely short-ranged, and so the binding necessitates matching of complementary shapes, i.e. lock-and-key mechanism in three dimensions. Proteins are created whenever needed, based on the information present in DNA, and disintegrated once their function is over.

The identification of these tasks makes it easy to see why there are two languages and not just one. Memory needs long term stability, on the other hand fast execution of functions is desirable, and the two make different demands on the hardware involved. (The accuracy of a single language performing both the tasks would be limited, which is the likely reason why the RNA world, described later, did not last very long.) Indeed, our electronic

computers compute using electrical signals, but store the results on the disk using magnetic signals. The former encoding is suitable for fast processing, while the latter is suitable for long term storage. The two hardware languages fortunately correspond to the same binary software language, and are conveniently translated into each other by the laws of electromagnetism. In case of genetic information, the two hardware languages work in different dimensions—DNA is a linear chain while proteins are three dimensional structures—forcing the corresponding software languages also to be different and the translation machinery fairly complex.

We want to find the optimal languages for implementing the tasks of DNA/RNA and proteins. So we have to study what constraints are imposed on a language for minimisation of errors and minimisation of resources. Minimisation of errors inevitably leads to a digital language, having a set of clearly distinguishable building blocks with discrete operations. With non-overlapping signals, small fluctuations (say less than half the separation between the discrete values) are interpreted as noise and eliminated from the message by resetting the values, while large changes represent genuine change in meaning. The loss of intermediate values is not a drawback, as long as actual applications need only results with bounded errors. Minimisation of resources is achieved by using a small number of building blocks, with simple and quick operations. A versatile language is then obtained by arranging the building blocks together in as many different ways as possible.

In this optimisation exercise, the “minimal language”, i.e. the language with the smallest set of building blocks for a given task, has a unique status [Patel (2006a)]:

- It has the largest tolerance against errors, since the discrete variables are spread as far apart as possible in the available range of physical hardware properties.
- It has the smallest instruction set, since the number of possible transformations is automatically limited.
- It can function with high density of packing and quick operations, which more than make up for the increased depth of computation.
- It can avoid the need for translation, by using simple physical responses of the hardware.

The genetic languages are undoubtedly digital, and that has been crucial in producing evolution as we know it. Some tell-tale signatures are:

- Digital language helps in maintaining variation, while continuous variables would average out fluctuations.

- It is a curious fact that evolution is a consequence of a tiny error rate. With too many errors the organism will not be able to survive, but without mutations there will be no evolution.
- Even minimal changes in discrete genetic variables generate sizeable disruptions in the system, and they will be futile unless the system can tolerate them. Often a large number of trial variations are needed to find the right combinations, and having only a small number of discrete possibilities helps. Continuous variables produce gradual evolution, which appears on larger phenotypic scales when multiple sources contributing to a particular feature average out.
- With most of the trial variations getting rejected as being unproductive, digital variables give rise to punctuated evolution—sudden changes interspersed amongst long periods of stasis.

In the following sections, we investigate to what extent the digital genetic languages are minimal, i.e. we first deduce the minimal languages for the tasks of DNA/RNA and proteins, and then compare them to what the living organisms have opted for. A worthwhile bonus is that we gain useful clues about the simpler predecessors of the modern genetic languages.

#### 10.4 Understanding Proteins

Finding the minimal language for proteins is a straightforward problem in classical geometry [Patel (2002)]. The following is a rapid-fire summary of the analysis.

- *What is the purpose of the language of amino acids?*  
To form protein molecules of different shapes and sizes in three dimensions, and containing different chemical groups.
- *What is the minimal discrete geometry for designing three dimensional structures?*  
Simplicial tetrahedral geometry and the diamond lattice. Secondary protein structures, i.e.  $\alpha$ -helices,  $\beta$ -bends and  $\beta$ -sheets, fit quite well on the diamond lattice.
- *What are the best physical components to realise this geometry?*  
Covalently bonded carbon atoms, also  $N^+$  and  $H_2O$ . Silicon is far more abundant, but it cannot form aperiodic structures needed to encode a language. (In the graphite sheet arrangement, carbon also provides the simplicial geometry for two dimensional membrane patterns.)

- What is a convenient way to assemble these components in the desired three dimensional structures?

Synthesise one dimensional polypeptide chains, which carry knowledge about how to fold into three dimensional structures. The problem then simplifies to assembling one dimensional chains. (Note that images in our electronic computers are stored as folded sequences.)

- What are the elementary operations needed to fold a polypeptide chain on a diamond lattice, in any desired manner?

Nine discrete rotations, represented as a  $3 \times 3$  array on the Ramachandran map (see Fig.10.2). Additional folding operations are trans-cis flip and long distance bonds.

- What can the side groups of polypeptide chains do?

They favour particular orientations of the polypeptide chain by interactions amongst themselves. They also fill up cavities in the structure by variations in their size.

To put the above statements in biological perspective, and to illustrate the minimalistic choices made by the living organisms (in the context of what was available), here are some facts about the polypeptide chains.

- Amino acids are easily produced in primordial chemical soup. They even exist in interstellar clouds.
- Amino acids are the smallest organic molecules with both an acid group ( $-COOH$ ) and a base group ( $-NH_2$ ). They differ from each other in terms of distinct R-groups, which become the side groups of polypeptide chains.
- Polypeptide chains are produced by polymerisation of amino acids by acid-base neutralisation (see Fig.10.1).
- Folded  $\leftrightarrow$  unfolded transition of polypeptide chains requires flexible joints and weak non-local interactions (close to critical behaviour).
- Transport of polypeptides across membranes is efficient in the unfolded

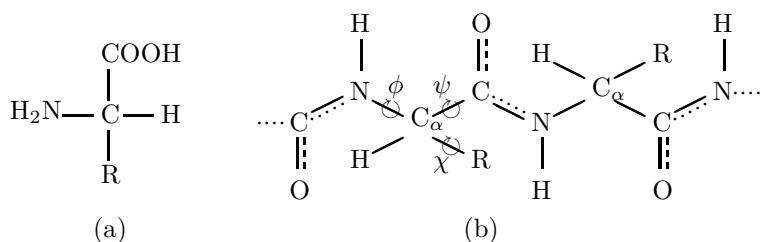


Fig. 10.1 Chemical structures of (a) amino acid, (b) polypeptide chain.

state than in the folded one, preventing leakage of other molecules at the same time. (A chain can slide through a small hole.)

The structural language of the polypeptide chains would be the most versatile when all possible orientations can be generated by every amino acid segment. This cannot be achieved by just a single property of the R-groups (e.g. hydrophobic to hydrophilic variation). The table below lists the amino acids used by the universal language of proteins. They are subdivided into several categories according to the chemical properties of the R-groups, and their molecular weights provide an indication of the size of the R-groups [Lehnninger *et al.* (1993)]. The language of bends and folds of the polypeptide chains is non-local, i.e. the orientation of an amino acid is not determined by its own R-group alone, rather the orientation is decided by the interactions of the amino acid with all its neighbours.

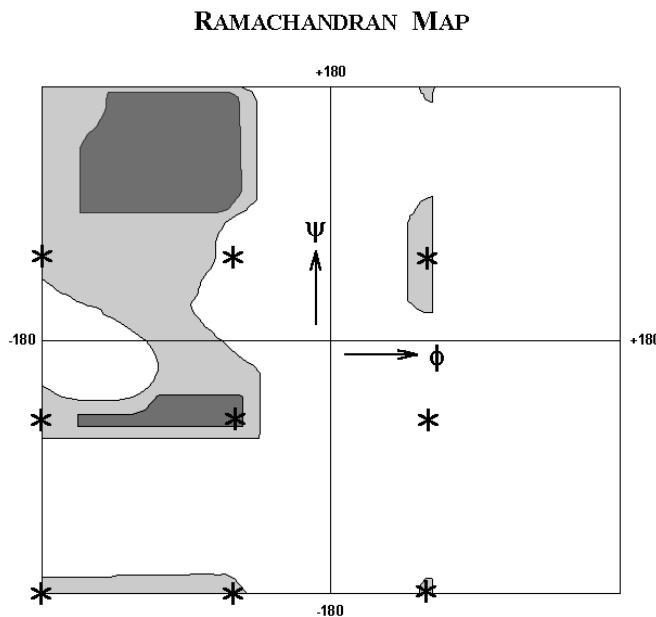


Fig. 10.2 The allowed orientation angles for the  $C_\alpha$  bonds in real polypeptide chains for chiral L-type amino acids, taking into account hard core repulsion between atoms [Ramachandran *et al.* (1963)]. Stars mark the nine discrete possibilities for the angles, uniformly separated by  $120^\circ$  intervals, when the polypeptide chain is folded on a diamond lattice.

Still, by analysing protein databases, one can find probabilities for every amino acid to participate in specific secondary structures, and the dominant propensities are listed in the table as well [Creighton (1993)].

Amino acid	R-group	Mol. wt.	Class	Propensity
G Gly (Glycine)	Non-polar	75	II	turn
A Ala (Alanine)	aliphatic	89	II	$\alpha$
P Pro (Proline)		115	II	turn
V Val (Valine)		117	I	$\beta$
L Leu (Leucine)		131	I	$\alpha$
I Ile (Isoleucine)		131	I	$\beta$
S Ser (Serine)	Polar	105	II	turn
T Thr (Threonine)	uncharged	119	II	$\beta$
N Asn (Asparagine)		132	II	turn
C Cys (Cysteine)		121	I	$\beta$
M Met (Methionine)		149	I	$\alpha$
Q Gln (Glutamine)		146	I	$\alpha$
D Asp (Aspartate)	Negative	133	II	turn
E Glu (Glutamate)	charge	147	I	$\alpha$
K Lys (Lysine)	Positive	146	II	$\alpha$
R Arg (Arginine)	charge	174	I	$\alpha$
H His (Histidine)	Ring/	155	II	$\alpha$
F Phe (Phenylalanine)	aromatic	165	II	$\beta$
Y Tyr (Tyrosine)		181	I	$\beta$
W Trp (Tryptophan)		204	I	$\beta$

Deciphering the actual orientations of amino acids in proteins is an outstanding open problem—the protein folding problem. Even then a rough count of the number of amino acids present can be obtained with one additional input. This is the division of the amino acids into two classes, according to the properties of the corresponding aminoacyl-tRNA synthetases (aaRS). In the synthesis of polypeptide chains, tRNA molecules are the adaptors with one end matching with a genetic codon and the other end attached to an amino acid. The aaRS are the truly bilingual molecules in the translation machinery, that attach an appropriate amino acid to the tRNA corresponding to its anticodon. There is a unique aaRS for every amino acid, even though several different tRNA molecules can carry the same amino acid (the genetic code is degenerate). It has been discovered that the aaRS are clearly divided in two classes, according to their

sequence and structural motifs, active sites and the location where they attach the amino acids to the tRNA molecules [Arnez and Moras (1997); Lewin (2000)]. The classes of amino acids are also listed in the table above, and here is what we find:

- (a) The 20 amino acids are divided into two classes of 10 each.
- (b) The two classes divide amino acids with each R-group property equally, in such a way that for every R-group property the larger R-groups correspond to class I and the smaller ones to class II.
- (c) The class label of an amino acid can be interpreted as a binary code for its R-group size, in addition to the categorisation in terms of chemical properties.
- (d) This binary code has unambiguous structural significance for packing of proteins. Folding of an aperiodic chain into a compact structure invariably leaves behind cavities of different shapes and sizes. The use of large R-groups to fill big cavities and small R-groups to fill small ones can produce dense compact structures.
- (e) Each class contains a special amino acid, involved in operations other than local folding of polypeptide chains—Cys in class I can make long distance disulfide bonds, and Pro in class II can induce trans-cis flip.

We thus arrive at a structural explanation for the 20 amino acids as building blocks of proteins. Local orientations of the polypeptide chains have to cover the nine discrete points on the Ramachandran map. They are governed by the chemical properties of the amino acid R-groups, and an efficient encoding can do the job with nine amino acids. The binary code for the R-group sizes fills up the cavities nicely without disturbing the folds. And then two more non-local operations increase the stability of protein molecules.

The above counting doesn't tell which sequence of amino acids will lead to which conformation of the polypeptide chain. That remains an unsolved exercise in coding as well as chemical properties. On the other hand, it is known that amino acids located at the active sites and at the end-points of secondary structures determine the domains and activity of proteins, while the amino acids in the intervening regions more or less act like space-filters. Among the space-filters, many substitutions can be carried out that hardly affect the protein function—indeed protein database analyses have produced probabilistic substitution tables for the amino acids. We need to somehow incorporate this feature into our understanding of the structural language of proteins, so that we can progress beyond individual letters to words and sentences [see for example, Socolich *et al.* (2005); Russ *et al.*

(2005)]. A new perspective is necessary, and perhaps the following self-explanatory paragraph is a clue [Rawlinson (1976)]. Surprise yourself by reading it at full speed, even if you are not familiar with crossword puzzles!

You arne't ginog to blveiee taht you can aulaclty uesdnatnrd waht I am wirtnig. Beuacse of the phaonmneal pweor of the hmuan mnid, aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoatnt tihng is taht the frist and lsat ltteer be in the rghit pclae. The rset can be a taotl mses and you can sitll raed it wouthit a porbelm. Tihs is bcuseae the huann mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe. Amzanig huh? Yaeh and you awlyas tghuhot slpelng was ipmorant!

Written English and proteins are both non-local languages. Evolution, after all, is no stranger to using a worthwhile idea—here a certain amount of parallel and distributed processing—over and over again.

## 10.5 Understanding DNA

Now let us move on to finding the minimal language for DNA and RNA. Once again, here is a quick-fire summary of the analysis [Patel (2001a)].

- *What is the information processing task carried out by DNA?*

Sequential assembly of a complementary copy on top of the pre-existing template by picking up single nucleotide bases from an unsorted ensemble. The same task is carried out by mRNA in the assembly of polypeptide chains, but proceeding in steps of three nucleotide bases (triplet codons).

- *What is the optimal way of carrying out this task?*

Lov Grover's database search algorithm [Grover (1996)], which uses binary queries and requires wave dynamics. It optimises the number of queries, providing a quadratic speed up over any Boolean algorithm, irrespective of the size of spatial resources the Boolean algorithm may use. In a classical wave implementation the database is encoded as  $N$  distinct wave modes, while in a quantum setting the database is labeled by  $\log_2 N$  qubits.

- *What is the characteristic signature of this algorithm?*

The number of queries  $Q$  required to pick the desired object from an

unsorted database of size  $N$  are given by:

$$(2Q + 1) \sin^{-1} \frac{1}{\sqrt{N}} = \frac{\pi}{2} \implies \begin{cases} Q = 1, & N = 4 \\ Q = 2, & N = 10.5 \\ Q = 3, & N = 20.2 \end{cases} \quad (10.1)$$

(Non-integral values of  $N$  imply small errors in object identification, about 1 part in 700 and 1050 for  $Q = 2$  and 3 respectively.)

- *What are the physical ingredients needed to implement this algorithm?*  
A system of coupled wave modes whose superposition maintains phase coherence, and two reflection operations (phase changes of  $\pi$ ).

Again to clarify the biological perspective, and to illustrate the minimalist choices made by the living organisms, here are some facts about the biochemical assembly process.

- (a) Instead of waiting for a desired complex biomolecule to come along, it is far more efficient to synthesise it from common, simple ingredients.
- (b) There should be a sufficient number of clearly distinguishable building blocks to create the wide variety of required biomolecules.
- (c) The building blocks are randomly floating around in the cellular environment. They get picked one by one and added to a linearly growing polymer chain.
- (d) Complementary nucleotide base-pairing decides the correct building block to be added at each step of the assembly process.
- (e) The base-pairings are binary questions; either they form or they do not form. The molecular bonds involved are hydrogen bonds.

With these features, the optimal classical algorithm based on Boolean logic would be a binary tree search. But the observed numbers do not fit that pattern (of powers of two). On the other hand, the optimal search solutions of Grover's algorithm are clearly different from and superior to the Boolean ones, and they do produce the right numbers. The crucial difference between the two is that wave mechanics works with amplitudes and not probabilities, which allows constructive as well as destructive interference. Grover's algorithm manages the interference of amplitudes cleverly, and the individual steps are depicted in Fig.10.3 for the simplest case of four items in the database.

Now note that classically the binary alphabet is the minimal one for encoding information in a linear chain, and two nucleotide bases (one complementary pair) are sufficient to encode the genetic information. As a

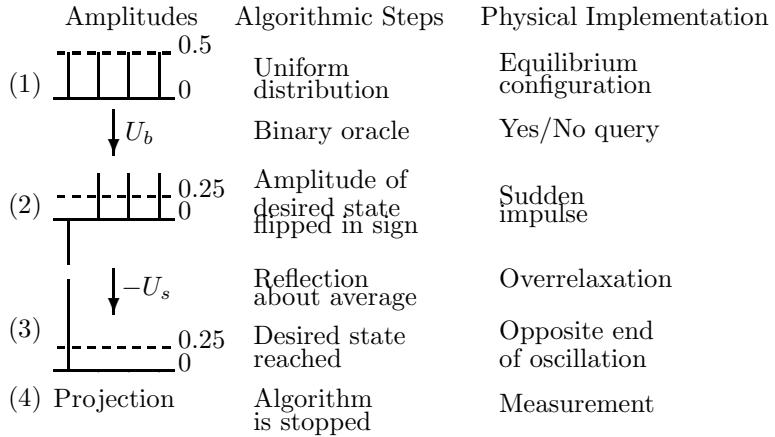


Fig. 10.3 The steps of Grover's database search algorithm for the simplest case of four items, when the first item is desired by the oracle. The left column depicts the amplitudes of the four states, with the dashed lines showing their average values. The middle column describes the algorithmic steps, and the right column mentions their physical implementation.

matter of fact, our digital computers encode all types of information using only 0's and 1's. The binary alphabet is the simplest system, and so would have preceded (during evolution) the four nucleotide base system found in nature. Then, was the speed-up provided by the wave algorithm the real incentive for nature to complicate the genetic alphabet? Certainly, if we have to design the optimal system for linear assembly, knowing all the physical laws that we do, we would opt for something like what is present in nature. But what did nature really do? We have no choice but to face the following questions:

- Does the genetic machinery have the ingredients to implement Grover's algorithm?

The physical components are definitely present, and it is not too difficult to construct scenarios based on quantum dynamics [Patel (2001a)] as well as vibrational motion [Patel (2006b)]. Although Grover's algorithm was discovered in the context of quantum computation, it is much more general, and does not need all the properties of quantum dynamics. In particular, highly fragile entanglement is unnecessary, while much more stable superposition of states is a must. The issue of concern then is whether coherent superposition of wave modes can sur-

vive long enough for the algorithm to execute. This superposition may be quantum (i.e. for the wavefunction) or may be classical (as in case of vibrations). It need not be exactly synchronous either—if the system transits through all the possible states at a rate much faster than the time scale of the selection oracle, that would simulate superposition, averaging out high frequency components (e.g. the appearance of spokes of a rapidly spinning wheel). Provided that the superposition is achieved somehow, the mathematical signature, i.e. Eq.(10.1), follows. Explicit formulation of a testable scenario, based on physical properties of the available molecules and capable of avoiding fast decoherence, is an open challenge.

- *Did nature actually exploit Grover’s algorithm when the genetic machinery evolved billions of years ago?*

Unfortunately there is no direct answer, since evolution of life cannot be repeated.

- *Do the living organisms use Grover’s algorithm even today?*

In principle, this is experimentally testable. Our technology is yet to reach a stage where we can directly observe molecular dynamics in a liquid environment. But indirect tests of optimality are plausible, e.g. constructing artificial genetic texts containing a different number of letters and letting it compete with the supposedly optimal natural language [Patel (2001b)].

This is not the end of the road, and I return to a deeper analysis later on. But prior to that let us look at what the above described understanding of the languages of proteins and DNA has to say about the translation mechanism between the two, i.e. the genetic code. That investigation does offer non-trivial rewards, regarding how the complex genetic machinery could have arisen from simpler predecessors.

## 10.6 What Preceded the Optimal Languages?

Languages of twenty amino acids and four nucleotide bases are too complex to be established in one go, and evolution must have arrived at them from simpler predecessors. On the other hand, continuity of knowledge has to be maintained in evolution from simpler to complex languages, because sudden drastic changes lead to misinterpretations that kill living organisms. Two evolutionary routes obeying this restriction, and still capable of producing large jumps, are known:

- (1) Duplication of information, which allows one copy to carry on the required function while the other is free to mutate and give rise to a new function.
  - (2) Wholesale import of fully functional components by a living organism, distinct from their own and developed by a different living organism.
- In what follows, we study the genetic languages within this framework.

The two classes of amino acids and the  $Q = 2$  solution of Grover's algorithm, described in preceding sections, suggest a duplication event, i.e. the universal non-overlapping triplet genetic code arose from a more primitive doublet genetic code labeling ten amino acids [Patel (2005); Wu *et al.* (2005); Rodin and Rodin (2006)]. To justify this hypothesis, we have to identify evolutionary remnants of (a) a genetic language where only two nucleotide bases of a codon carry information while the third one is a punctuation mark, (b) a set of amino acids that can produce all the orientations of polypeptide chains but without efficiently filling up the cavities, and (c) a reasonable association between these codons and amino acids. Amazingly, biochemical signals for all of these features have been observed.

The central players in this event are the tRNA molecules. They are older than the DNA and the proteins in evolutionary history, and are believed to link the modern genetic machinery with the earlier RNA world [Gesteland *et al.* (2006)]. It has been discovered that RNA polymers called ribozymes can both store information and function as catalytic enzymes, although not very accurately. The hypothesis is that when more accurate DNA and proteins took over these tasks from ribozymes, tRNA molecules

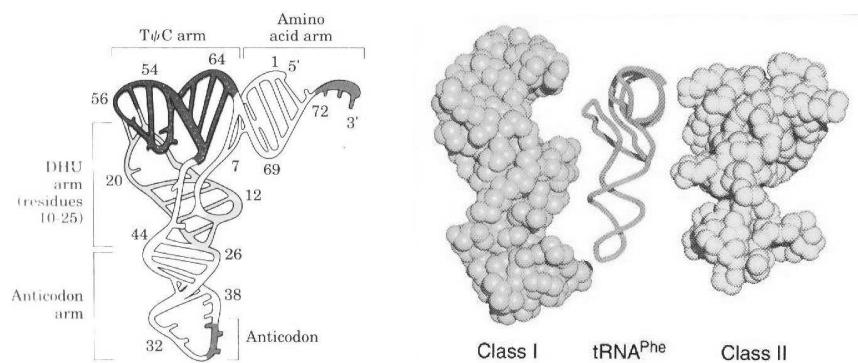


Fig. 10.4 The structure of tRNA [Lehninger *et al.* (1993)] (left), and the tRNA-AARS interaction from opposite sides for the two classes [Arnez and Moras (1997)] (right).

survived as adaptors from the preceding era.

As illustrated in Fig.10.4, the tRNAs are L-shaped molecules with the amino acid acceptor arm at one end and the anticodon arm at the other. The two arms are separated by a distance of about 75Å, too far apart for any direct interaction. The aaRS molecules are much larger than the tRNAs, and they attach an amino acid to the acceptor stem corresponding to the anticodon by interacting with both the arms. The two classes of aaRS perform this attachment from opposite sides, in a mirror image fashion as shown in Fig.10.4. Class I attachment is from the minor groove side of the acceptor arm helix, and class II attachment is from the major groove side. It has been observed that the tRNA acceptor stem sequence, which directly interacts with the R-group of the amino acid being attached, plays a dominant role in the amino acid recognition and the anticodon does not matter much. This behaviour characterises the operational RNA code, formed by the first four base pairs and the unpaired base N<sup>73</sup> of the acceptor stem [Schimmel *et al.* (1993)]. The operational code relies on stereochemical atomic recognition between amino acid R-groups and nucleotide bases; it is argued to be older than the genetic code and a key to understanding the goings on in the RNA world.

The universal genetic code

<b>UUU Phe</b>	<b>UCU Ser</b>	<b>UAU Tyr</b>	<b>UGU Cys</b>
<b>UUC Phe</b>	<b>UCC Ser</b>	<b>UAC Tyr</b>	<b>UGC Cys</b>
<b>UUA Leu</b>	<b>UCA Ser</b>	<b>UAA Stop</b>	<b>UGA Stop</b>
<b>UUG Leu</b>	<b>UCG Ser</b>	<b>UAG Stop</b>	<b>UGG Trp</b>
<b>CUU Leu</b>	<b>CCU Pro</b>	<b>CAU His</b>	<b>CGU Arg</b>
<b>CUC Leu</b>	<b>CCC Pro</b>	<b>CAC His</b>	<b>CGC Arg</b>
<b>CUA Leu</b>	<b>CCA Pro</b>	<b>CAA Gln</b>	<b>CGA Arg</b>
<b>CUG Leu</b>	<b>CCG Pro</b>	<b>CAG Gln</b>	<b>CGG Arg</b>
<b>AUU Ile</b>	<b>ACU Thr</b>	<b>AAU Asn</b>	<b>AGU Ser</b>
<b>AUC Ile</b>	<b>ACC Thr</b>	<b>AAC Asn</b>	<b>AGC Ser</b>
<b>AUA Ile</b>	<b>ACA Thr</b>	<b>AAA Lys</b>	<b>AGA Arg</b>
<b>AUG Met</b>	<b>ACG Thr</b>	<b>AAG Lys</b>	<b>AGG Arg</b>
<b>GUU Val</b>	<b>GCU Ala</b>	<b>GAU Asp</b>	<b>GGU Gly</b>
<b>GUC Val</b>	<b>GCC Ala</b>	<b>GAC Asp</b>	<b>GGC Gly</b>
<b>GUU Val</b>	<b>GCA Ala</b>	<b>GAA Glu</b>	<b>GGA Gly</b>
<b>GUG Val</b>	<b>GCG Ala</b>	<b>GAG Glu</b>	<b>GGG Gly</b>

Boldface letters indicate class II amino acids.

We now look at the amino acid class pattern in the genetic code. The universal triplet genetic code has considerable and non-uniform degeneracy, with 64 codons carrying 21 signals (including Stop) as shown. Although there is a rough rule of similar codons for similar amino acids, no clear pattern is obvious.

By analysing genomes of living organisms, it has been found that during the translation process 61 mRNA codons (excluding Stop) pair with a smaller number of tRNA anticodons. The smaller degeneracy of the anticodons is due to wobble pairing of nucleotide bases, where the third base carries only a limited meaning (either binary or none) instead of four-fold possibilities [Crick (1966)]. The wobble rules are exact for the mitochondrial code—all that matters is whether the third base is a purine or a pyrimidine, and the number of possibilities reduces to 32 as shown. (Note that the mitochondrial code works with rather small genomes and evolves faster than the universal code, and so is likely to have simpler optimisation criteria.)

The (vertebrate) mitochondrial genetic code

<b>UUY Phe</b>	<b>UCY Ser</b>	UAY Tyr	UGY Cys
UUR Leu	<b>UCR Ser</b>	UAR Stop	UGR Trp
CUY Leu	<b>CCY Pro</b>	<b>CAY His</b>	CGY Arg
CUR Leu	<b>CCR Pro</b>	CAR Gln	CGR Arg
AUY Ile	<b>ACY Thr</b>	<b>AAY Asn</b>	<b>AGY Ser</b>
AUR Met	<b>ACR Thr</b>	<b>AAR Lys</b>	AGR Stop
GUY Val	<b>GCY Ala</b>	<b>GAY Asp</b>	<b>GGY Gly</b>
GUR Val	<b>GCR Ala</b>	GAR Glu	<b>GGR Gly</b>

Boldface letters indicate class II amino acids.

Pyrimidines Y=U or C, Purines R=A or G.

The departures exhibited by the mitochondrial genetic code, as well as the genetic codes of some living organisms, from the universal genetic code are rather minor, and only occur in some of the positions occupied by class I amino acids. It can be seen that all the class II amino acids, except Lys, can be coded by codons NNY and anticodons  $\overleftarrow{GNN}$  (wobble rules allow pairing of G with both U and C) [Patel (2005)]. This pattern suggests that the structurally more complex class I amino acids entered the genetic machinery later, and a doublet code for the class II amino acids (with the third base acting only as a punctuation mark) preceded the universal genetic code.

The class pattern becomes especially clear with two more inputs:

- (1) According to the sequence and structural motifs of their aaRS, Phe is assigned to class I and Tyr to class II. But if one looks at the stereochemistry of how the aaRS attach the amino acid to tRNA, then Phe belongs to class II and Tyr to class I [Goldgur *et al.* (1997); Yaremchuk *et al.* (2002)]. Thus from the operational RNA code point of view the two need to be swapped.
- (2) Lys has two distinct aaRS, one belonging to class I (in most archaea) and the other belonging to class II (in most bacteria and all eukaryotes) [Woese *et al.* (2000)]. On the other hand, the assignment of AGR codons varies from Arg to Stop, Ser and Gly. This feature is indicative of an exchange of class roles between AAR and AGR codons (models swapping Lys and Arg through ornithine have been proposed).

These two swaps of class labels do not alter the earlier observation that the two amino acid classes divide each R-group property equally. We thus arrive at the predecessor genetic code shown below. The binary division of the codons according to the class label is now not only unmistakable but produces a perfect complementary pattern [Rodin and Rodin (2006)].

The predecessor genetic code			
UUY Phe	<b>UCY Ser</b>	<b>UAY Tyr</b>	UGY Cys
UUR Leu	<b>UCR Ser</b>	UAR Stop	UGR Trp
CUY Leu	<b>CCY Pro</b>	<b>CAY His</b>	CGY Arg
CUR Leu	<b>CCR Pro</b>	CAR Gln	CGR Arg
AUY Ile	<b>ACY Thr</b>	<b>AAY Asn</b>	<b>AGY Ser</b>
AUR Met	<b>ACR Thr</b>	AAR Lys*	<b>AGR Arg*</b>
GUY Val	<b>GCY Ala</b>	<b>GAY Asp</b>	GGY Gly
GUR Val	<b>GCR Ala</b>	GAR Glu	<b>GGR Gly</b>

Boldface letters indicate class II amino acids.

Pyrimidines Y=U or C, Purines R=A or G.

When the middle base is Y (the first two columns), it indicates the class on its own—U for class I and C for class II. When the middle base is R (the last two columns), the class is denoted by an additional Y or R, in the third position when the middle base is A and in the first position when the middle base is G. (Explicitly the class I codons are NUN, NAR and YGN, while the class II codons are NCN, NAY and RGN.) The feature that after the middle base, the first or the third base determines the amino acid class in a complementary pattern, has led to the hypothesis that the amino acid class doubling occurred in a strand symmetric RNA world, with complementary

tRNAs providing complementary anticodons. [Rodin and Rodin (2006)].

The complementary pattern has an echo in the operational code of the tRNA acceptor stem too. When the aaRS attach the amino acid to the –CCA tail of the tRNA acceptor arm, the tail bends back scorpion-like, and the R-group of the amino acid gets sandwiched between the tRNA acceptor stem groove (bases 1-3 and 70-73) and the aaRS. Analysis of tRNA consensus sequences from many living organisms reveals [Rodin and Rodin (2006)] that (a) the first base pair in the acceptor stem groove is almost invariably G<sup>1</sup>-C<sup>72</sup> and is mapped to the wobble position of the codon, (b) the second base pair is mostly G<sup>2</sup>-C<sup>71</sup> or C<sup>2</sup>-G<sup>71</sup>, which correlate well respectively with Y and R in the middle position of the codon, and (c) the other bases do not show any class complementarity pattern.

The involvement of both the operational RNA code and the anticodon in the selection of appropriate amino acid, and the above mentioned correlations between the two, make it very likely that the two had a common origin. Then piecing together all the observed features, the following scenario emerges for the evolution of the genetic code:

(1) Ribozymes of the RNA world could replicate, but their functional capability was limited—a small alphabet (quite likely four nucleotide bases) and restricted conformations could only produce certain types of structures. Polypeptide chains, even with a small repertoire of amino acids, provided a much more accurate and versatile structural language, and they took over the functional tasks from ribozymes. This takeover required close stereochemical matching between ribozymes and polypeptide chains, in order to retain the functionalities already developed.

(2) The class II amino acids provided (or at least dominated) the initial structural language of proteins. With smaller R-groups, they are easier to synthesise, and so are likely to have appeared earlier in evolution. They can fold polypeptide chains in all possible conformations, although some of the cavities may remain incompletely filled. They also fit snugly into the major groove of the tRNA acceptor stem, with the bases 1-3 and 70-74 essentially forming a mould for the R-group, for precise stereochemical recognition. Indeed, this stereochemical identification of an R-group by three base pairs, necessitated by actual sizes of molecules, would be the reason for the triplet genetic code, even in a situation where all the bases do not carry information.

(3) The modern tRNA molecules arose from repetitive extensions and complementary pairing of short acceptor stem sequences. In the process, the

1-2-3 bases became the forerunners of the 34-35-36 anticodons. With different structural features identifying the amino acids, paired bases in the acceptor stem and unpaired bases in the anticodon, the evolution of the operational code and the genetic code diverged. The two are now different in exact base sequences, but the purine-pyrimidine label (i.e. R vs. Y) still shows high degree of correlation between the two.

(4) In the earlier era of class II amino acid language, the wobble base was a punctuation mark (likely to be G in the anticodon, as descendant of the 1-72 pair), the central base was the dominant identifier (descendant of the 2-71 pair), and the last anticodon base provided additional specification (equivalent to the 3-70 pair and the unpaired base 73). During subsequent evolution, these  $\overleftarrow{\text{GNN}}$  anticodons have retained their meaning, and all minor variations observed between genetic codes are in the other anticodons corresponding to class I amino acids.

(5) Class I amino acids got drafted into the structural language, because they could increase stability of proteins by improved packing of large cavities without disrupting established structures. The required binary label for the R-group size, appeared differently in the operational code and the genetic code. For the operational code, the minor groove of the acceptor stem was used, and utilisation of the same paired bases from the opposite side led to a complementary pattern. The class I amino acids fit loosely in the minor groove, and subsequent proof-reading is necessary at times to remove incorrectly attached amino acids. For the genetic code, several of the unassigned anticodons were used for the class I amino acids, introducing a binary meaning to the wobble position whenever needed. The Darwinian selection constraint that the operational code and the genetic code serve a common purpose ensured a rough complementary strand symmetry for the anticodons as well.

(6) The structural language reached its optimal stage, once both classes of amino acids were incorporated. With 32 anticodons (counting only a binary meaning for the wobble position) and 20 amino acid signals, enough anticodons may have remained unassigned. Most of them were taken over by amino acids with close chemical affinities (wobble position did not assume any meaning), and a few left over ones mapped to the Stop signal.

(7) All this could have happened when each gene was a separate molecule, coding for a single polypeptide chain. Additional selection pressures must have arisen when the genes combined into a genome. To take care of the increased complexity, some juggling of codons happened and the Start signal appeared. The present analysis is not detailed enough to explain this

later optimisation. Nevertheless, interpretation of similar codons for similar amino acids and the wobble rules, as relics of the doubling of the genetic code—indicative but not perfect—is a significant achievement.

At the heart of the class duplication mechanism described above is (a) the mirror image pattern of the amino acid R-group fit with the tRNA acceptor stem, and (b) the complementary pattern of the anticodons. More detailed checks for these are certainly possible. The amino acids have been tested for direct chemical affinities with either their codons or their anticodons (but not both together), and most results have been lukewarm [Yarus *et al.* (2005)]. Instead, chemical affinities of amino acids with paired codon-anticodon grooves should be tested, both by stereochemical models and actual experiments. It should be also possible to identify which amino acid paired with which one when the genetic code doubled. Some pairs can be easily inferred from biochemical properties [Ribas de Pouplana and Schimmel (2001); Patel (2005)]—(Asp,Glu), (Asn,Gln), (Lys,Arg), (Pro,Cys), (Phe,Tyr), (Ser&Thr,Val&Ile)—while the others would be revealed by stereochemical modeling.

The next interesting exercise, further back in time and therefore more speculative, is to identify how a single class 10 amino acid language took over the functional tasks of 4 nucleotide base ribozymes. This is the stage where Grover’s algorithm might have played a crucial role, and so we go back and look into it more inquisitively.

### **10.7 Quantum Role?**

The arguments of the preceding section reduce the amino acid identification problem by a triplet code, to the identification problem within a class by a doublet code plus a binary class label. It is an accidental degeneracy that the  $Q = 3$  solution of Grover’s algorithm, Eq.(10.1), can be obtained as the  $Q = 2$  solution plus a classical binary query. To assert that the sequential assembly process reached its optimal solution, we still need to resolve how the  $Q = 1, 2$  solutions of Eq.(10.1) were realised by the primordial living organisms.

Clearly, the assembly processes occur at the molecular scale. We know the physical laws applicable there—classical dynamics is relevant, but quantum dynamics cannot be bypassed. Discrete atomic structure provided by quantum mechanics is the basis of digital genetic languages. Molecular bonds are generally given a classical description, but they cannot take place

without appropriate quantum correlations among the electron wavefunctions. Especially, hydrogen bonds are critical to the genetic identification process, and they are inherently quantum—typical examples of tunneling in a double well potential. The assemblers, i.e. the polymerase enzymes and the aaRS molecules, are much larger than the nucleotide bases and the amino acids, and completely enclose the active regions where identification of nucleotide bases and amino acids occurs. They provide a well-shielded environment for the assembly process, but the cover-up also makes it difficult to figure out what exactly goes on inside.

Chemical reactions are typically described in terms of specific initial and final states, and transition matrix elements between the two that characterise the reaction rates. That is a fully classical description, and it works well for most practical purposes. But to the best of our understanding, the fundamental laws of physics are quantum and not classical—the classical behaviour arises from the quantum world as an “averaged out” description. Quantum steps are thus necessarily present inside averaged out chemical reaction rates, and would be revealed if we can locate their characteristic signatures. In the present context, such a fingerprint is superposition.

The initial and final states of Grover’s algorithm are classical, but the execution in between is not. In order to be stable, the initial and final states have to be based on a relaxation towards equilibrium process. For the execution of the algorithm in between, the minimal physical requirement is a system that allows superposition of states, in particular a set of coupled wave modes. As illustrated earlier in Fig.10.3, the algorithm needs two reflection operations. Provided that the necessary superposition is achieved somehow, it is straightforward to map these operations to: (i) the impulse interaction during molecular bond formation which has the right properties to realise the selection oracle as a fairly stable geometric phase, and (ii) the (damped) oscillations of the subsequent relaxation, which when stopped at the right instant by release of the binding energy to the environment can make up the other reflection phase.

Beyond this generic description, the specific wave modes to be superposed can come from a variety of physical resources, e.g. quantum evolution, vibrations and rotations. With properly tuned couplings, resonant transfer of amplitudes occurs amongst the wave modes (the phenomenon of beats), and that is the dynamics of Grover’s algorithm. When the waves remain coherent, their amplitudes add and subtract, and we have superposition. But when the waves lose their coherence, we get an averaged out result—a classical mixture. Thus the bottom line of the problem is:

*Can the genetic machinery maintain coherence of appropriate wave modes on a time scale required by the transition matrix elements?*

Explicitly, let  $t_b$  be the time for molecular identification by bond formation,  $t_{coh}$  be the time over which coherent superposition holds, and  $t_{rel}$  be the time scale for relaxation to equilibrium. Then, Grover's algorithm can be executed when the time scales satisfy the hierarchy

$$t_b \ll t_{coh} \ll t_{rel} . \quad (10.2)$$

Other than this constraint, the algorithm is quite robust and does not rely on fine-tuned parameters. (Damping is the dominant source of error; other effects produce errors which are quadratic in perturbation parameters.)

Wave modes inevitably decohere due to their interaction with environment, essentially through molecular collisions and long range forces. Decoherence always produces a cross-over leading to irreversible loss of information [Guilini *et al.* (1996)]—collapse of the wavefunction in the quantum case and damped oscillations for classical waves. The time scales of decoherence depend on the dynamics involved, but a generic feature is that no wave motion can be damped faster than its natural undamped frequency of oscillation. For an oscillator,

$$\ddot{x} + 2\gamma\dot{x} + \omega_0^2 x = 0, \quad x \sim e^{i\omega t} \implies \gamma_{\text{crit}} = \max(\text{Im}(\omega)) = \omega_0 . \quad (10.3)$$

Too much damping freezes the wave amplitude instead of making it decay. Thus  $\omega_0^{-1}$  is both an estimate of  $t_b$  and a lower bound on  $t_{coh}$ . Molecular properties yield  $\omega_0 = \Delta E/\hbar = O(10^{14})\text{sec}^{-1}$ , for the transition frequencies of weak bonds as well as for the vibration frequencies of covalent bonds.

Decoherence must be controlled in order to observe wave dynamics, irrespective of any other (undiscovered) physical phenomena that may be involved. In case of vibrational and rotational modes of molecules, the fact that we can experimentally measure the excitation spectra implies that the decoherence times are much longer than  $t_b$ . In case of quantum dynamics, the decoherence rate is often estimated from the scattering cross-sections of environmental interactions, in dilute gas approximation using conventional thermodynamics and Fermi's golden rule. For molecular processes, these times are usually minuscule, orders of magnitude below  $\omega_0^{-1}$ . In view of Eq.(10.3), such minuscule estimates are wrong—the reason being that Fermi's golden rule is an approximation, not valid at times smaller than the natural oscillation period. A more careful analysis is necessary.

According to Fermi's golden rule, the environmental decoherence rate is inversely proportional to three factors: the initial flux, the interaction strength and the final density of states. We know specific situations, where quantum states are long-lived due to suppression of one or more of these factors. The initial flux is typically reduced by low temperatures and shielding, the interaction strength is small for lasers and nuclear spins, and the final density of states is suppressed due to energy gap for superconductors and hydrogen bonds. We need to investigate whether or not these features are exploited by the genetic machinery, and if so to what extent.

Large catalytic enzymes (e.g. polymerases, aaRS, ribosomes) have an indispensable role in biomolecular assembly processes. These processes do not take place in thermal equilibrium, rather the enzymes provide an environment that supplies free energy (using ATP molecules) as well as shields. The assembly then proceeds along the chain linearly in time. In a free solution without the enzymes, the assembly just does not take place, even though such a free assembly would have the advantage of parallel processing (i.e. simultaneous assembly all along the chain). The enzymes certainly reduce the external disturbances and decrease the final density of states by limiting possible configurations. But much more than that, they stabilise the intermediate reaction states, called the transition states. The traditional description is that the free energy barrier between the reactants and the products is too high to cross with just the thermal fluctuations, and the enzymes take the process forward by lowering the barrier and supplying free energy. The transition states are generally depicted using distorted electron clouds, somewhere in between the configurations of the reactants and the products, and they are unstable when not assisted by the enzymes. They can only be interpreted as superpositions, and not as mixtures—we have to accept that the enzymes stabilise such intermediate superposition states while driving biomolecular processes.

Thus we arrive at the heart of the inquiry:

*Grover's algorithm needs certain type of superpositions, and catalytic enzymes can stabilize certain type of superpositions. Do the two match, and if so, what is the nature of this superposition?*

The specific details of the answer depend on the dynamical mechanism involved. The requisite superposition is of molecules that have a largely common structure while differing from each other by about 5-10 atoms. I have proposed two possibilities [Patel (2001a); Patel (2006b)]:

- (1) In a quantum scenario, wavefunctions get superposed and the algorithm

enhances the probability of finding the desired state. Chemically distinct molecules cannot be directly superposed, but they can be effectively superposed by a rapid cut-and-paste job of chemical groups (enzymes are known to perform such cut-and-paste jobs). Whether this really occurs, faster than the identification time scale  $t_b$  and with the decoherence time scale significantly longer than  $\hbar/\omega_0$ , is a question that should be experimentally addressed. It is a tough proposition, and most theoretical estimates are pessimistic.

(2) In a classical wave scenario, all the candidate molecules need to be present simultaneously and coupled together in a specific manner. The algorithm concentrates mechanical energy of the system into the desired molecule by coherent oscillations, helping it cross the energy barrier and complete the chemical reaction. Enzymes are required to couple the components together with specific normal modes of oscillation, and long enough coherence times are achievable. This scenario provides the same speed up in the number of queries  $Q$  as the quantum one, but involves extra spatial costs. The extra cost is not insurmountable in the small  $N$  solutions relevant to genetic languages, and the extra stability against decoherence makes the classical wave scenario preferable. (Once again note that time optimisation is far more important in biology than space optimisation.)

Twists and turns can be added to these scenarios while constructing a detailed picture. But in any implementation of Grover's algorithm, the requirement of superposition would manifest itself as simultaneous presence of all the candidate molecules during the selection process, in contrast to the one-by-one trials of a Boolean algorithm. This particular aspect can be experimentally tested by the available techniques of isotope substitution, NMR spectroscopy and resonance frequency measurements. The algorithm also requires the enzymes to play a central role in driving the non-equilibrium selection process, but direct observation of that would have to await breakthroughs in technologies at nanometre and femtosecond scales.

## 10.8 Outlook

Information theory provides a powerful framework for extracting essential features of complicated processes of life, and then analysing them in a systematic manner. The easiest processes to study are no doubt the ones at the lowest level. We have learned a lot, both in computer science and in molecular biology, since their early days [Schrödinger (1944);

von Neumann (1958); Crick (1968)], and so we can now perform a much more detailed study. Physical theories often start out as effective theories, where predictions of the theories depend on certain parameters. The values of the parameters have to be either assumed or taken from experiments; the effective theory cannot predict them. To understand why the parameters have the values they do, we have to go one level deeper—typically to smaller scales. When the deeper level reduces the number of unknown parameters, we consider the theory to be more complete and satisfactory. The level below conventional molecular biology is spanned by atomic structure and quantum dynamics, and that is the natural place to look for reasons behind life’s “frozen accident”. It is indeed wonderful that sufficient ingredients exist at this deeper level to explain the frozen accident as the optimal solution. The first reward of this analysis has been a glimpse of how the optimal solution was arrived at.

Evolution of life occurs through random events (i.e. mutations), without any foresight or precise rules of logic. It is the powerful criterion of survival, in a usually uncomfortable and at times hostile environment, that provides evolution a direction. Even though we do not really understand why living organisms want to perpetuate themselves, we have enough evidence to show that they use all available means for this purpose [Dawkins (1989)]. This struggle for fitness allows us to assign underlying patterns to evolution—not always perfect, frequently with variations, and yet very much practical. By understanding these patterns, we can narrow down the search for a likely evolutionary route among a multitude of possibilities. Such an insight is invaluable when we want to extrapolate in the unknown past with scant direct evidence. That is certainly the case in trying to understand the origin of life as we know it. Of course, the inferences become stronger when supported by simulated experiments, and worthwhile tests of every hypothesis presented have been pointed out in the course of this article.

Counting the number of building blocks in the languages of DNA and proteins, and finding patterns in them, is only the beginning of a long exercise to master these languages. Natural criteria for the selection of particular building blocks would be chemical simplicity (for easy availability and quick synthesis) and functional ability (for implementing the desired tasks). Life can be considered to have originated, not with just complex chemical interactions in a primordial soup, but only when the knowledge of functions of biomolecules started getting passed from one generation to the next. This logic puts the RNA world before the modern genetic ma-

achinery; ribozymes provide both function and memory, to a limited extent but with simpler ingredients. During evolution, the structurally more versatile polypeptides—they have been observed to successfully mimic DNA [Walkinshaw *et al.* (2002)] as well as tRNA [Nakamura (2001)]—took over the task of creating complex biochemistry, while leaving the memory storage job to DNA. The work described in this article definitely reinforces this point of view, with simpler predecessors of the modern genetic languages to be found in the stereochemical interaction between the tRNA acceptor stem and simple class II amino acids. Experimental verification of this hypothesis would by and large solve the translation mystery, i.e. which amino acid corresponds to which codon/anticodon. Then we can push the analysis further back in time, to the still simpler language of ribozymes, and try to figure out what went on in the RNA world.

The opposite direction of investigation, of constructing words and sentences from the letters of alphabets, is much more than a theoretical adventure and closely tied to what the future holds for us. We want to design biomolecules that carry out specific tasks, and that needs unraveling how the functions are encoded in the three dimensional protein assembly process. This is a tedious and difficult exercise, involving hierarchical structures and subjective variety. But some clues have appeared, and they should be built on to understand more and more complicated processes of life. We may feel uneasy and scared about consequences of redesigning ourselves, but that after all would also be an inevitable part of evolution!

*“...when the pie was opened, the birds began to sing ...”*

#### About the author

**Apoorva D. Patel** is a Professor at the Centre for High Energy Physics, Indian Institute of Science, Bangalore. He obtained his M.Sc. in Physics from the Indian Institute of Technology, Mumbai (1980), and Ph.D. in Physics from the California Institute of Technology (Caltech) under Geoffrey Fox (1984). His major field of work has been the theory of QCD, where he has used lattice gauge theory techniques to investigate spectral properties, phase transitions and matrix elements. In recent years, he has worked on quantum algorithms, and used information theory concepts to understand the structure of genetic languages.

## Bibliography

- Arnez, J.G. and Moras, D. (1997). Structural and Functional Considerations of the Aminoacyl Reaction, *Trends Biochem. Sci.* **22**, pp. 211-216.
- Aumann, R.J. (2006). War and Peace, *Nobel Prizes 2005: Les Prix Nobel*, Ed. K. Grandin (The Nobel Foundation, Stockholm), pp. 350-358.
- Creighton, T.E. (1993). *Proteins: Structures and Molecular Properties*, Second edition (W.H. Freeman, New York).
- Crick, F.H.C. (1966). Codon-Anticodon Pairing: The Wobble Hypothesis, *J. Mol. Biol.* **19**, pp. 548-555.
- Crick, F.H.C. (1968). The Origin of the Genetic Code, *J. Mol. Biol.* **38**, pp. 367-379.
- Dawkins, R. (1989). *The Selfish Gene* (Oxford University Press, Oxford).
- Gesteland, R.F., Cech, T.R. and Atkins, J.F. (Eds.) (2006). *The RNA World*, Third Edition (Cold Spring Harbor Laboratory Press, Cold Spring Harbor).
- Giulini, D., Joos, E., Kiefer, C., Kupsch, J., Stamatescu, I.-O. and Zeh, H.D. (1996). *Decoherence and the Appearance of a Classical World in Quantum Theory* (Springer, Berlin).
- Goldgur, Y., Mosyak L., Reshetnikova, L., Ankilova, V., Lavrik, O., Khodyreva, S. and Safro, M.G. (1997). The Crystal Structure of Phenylalanyl-tRNA Synthetase from Th. thermophilus Complexed with Cognate tRNA<sup>Phe</sup>, *Structure* **5**, pp. 59-68.
- Grover, L.K. (1996). A Fast Quantum Mechanical Algorithm for Database Search, *Proc. 28th Annual ACM Symposium on Theory of Computing*, Philadelphia, pp. 212-219, [arXiv.org:quant-ph/9605043](https://arxiv.org/abs/quant-ph/9605043).
- Lehninger, A.L., Nelson, D.L. and Cox, M.M. (1993). *Principles of Biochemistry*, Second edition (Worth Publishers, USA).
- Lewin, B. (2000). *Genes VII* (Oxford University Press, Oxford).
- Nakamura, Y. (2001). Molecular Mimicry Between Protein and tRNA, *J. Mol. Evol.* **53**, pp. 282-289.
- Patel, A. (2001a). Quantum Algorithms and the Genetic Code, *Pramana* **56**, pp. 367-381, [arXiv.org:quant-ph/0002037](https://arxiv.org/abs/quant-ph/0002037).
- Patel, A. (2001b). Testing Quantum Dynamics in Genetic Information Processing, *J. Genet.* **80**, pp. 39-43, [arXiv.org:quant-ph/0102034](https://arxiv.org/abs/quant-ph/0102034).
- Patel, A. (2002). Carbon—The First Frontier of Information Processing, *J. Biosc.* **27**, pp. 207-218, [arXiv.org:quant-ph/0103017](https://arxiv.org/abs/quant-ph/0103017).
- Patel, A. (2003). Mathematical Physics and Life, *Computing and Information Sciences: Recent Trends*, Ed. J.C. Misra (Narosa, New Delhi), pp. 271-294,

- [arXiv.org:quant-ph/0202022](https://arxiv.org/abs/quant-ph/0202022).
- Patel, A. (2005). The Triplet Genetic Code had a Doublet Predecessor, *J. Theor. Biol.* **233**, pp. 527-532, [arXiv.org:q-bio.GN/0403036](https://arxiv.org/abs/q-bio.GN/0403036).
- Patel, A. (2006a). The Future of Computation, *Proc. QICC 2005*, IIT Kharagpur, Ed. S.P. Pal and S. Kumar (Allied Publishers, Mumbai), pp. 197-206, [arXiv.org:quant-ph/0503068](https://arxiv.org/abs/quant-ph/0503068).
- Patel, A. (2006b). Optimal Database Search: Waves and Catalysis, *Int. J. Quant. Inform.* **4**, pp. 815-825; Erratum *ibid.* **5**, p. 437, [arXiv.org:quant-ph/0401154](https://arxiv.org/abs/quant-ph/0401154).
- Ramachandran, G.N., Ramakrishnan, C. and Sasisekharan, V. (1963). *J. Mol. Biol.* **7**, pp. 95-99.
- Rawlinson, G.E. (1976). The Significance of Letter Position in Word Recognition, *Ph.D. Thesis*, Psychology Department, University of Nottingham, UK.
- Ribas de Pouplana, L. and Schimmel, P. (2001). Aminoacyl-tRNA Synthetases: Potential Markers of Genetic Code Development, *Trends Biochem. Sci.* **26**, pp. 591-596.
- Rodin, S.N. and Rodin, A.S. (2006). Partitioning of aminoacyl-tRNA synthetases in two classes could have been encoded in a strand-symmetric RNA world, *DNA Cell Biol.* **25**, pp. 617-626.
- Russ, W.P., Lowery, D.M., Mishra, P., Yaffe, M.B. and Ranganathan, R. (2005). Nature-like Function in Artificial WW Domains, *Nature* **437**, pp. 579-583.
- Schimmel, P., Giege, R., Moras, D. and Yokoyama, S. (1993). An Operational RNA code for Amino Acids and Possible Relationship to Genetic Code, *Proc. Natl. Acad. Sci. USA* **90**, pp. 8763-8768.
- Schrödinger, E. (1944). *What is Life?* (Cambridge University Press, Cambridge).
- Shannon, C.E. (1948). A Mathematical Theory of Communication, *Bell System Tech. J.* **27**, pp. 379-423; 623-656.
- Socolich, M., Lockless, S.W., Russ, W.P., Lee, H., Gardner, K.H. and Ranganathan, R. (2005). Evolutionary Information for Specifying a Protein Fold, *Nature* **437**, pp. 512-518.
- von Neumann, J. (1958). *The Computer and the Brain* (Yale University Press, New Haven).
- Walkinshaw, M.D., Taylor, P., Sturrock, S.S., Atanasiu, C., Berge, T., Henderson, R.M., Edwardson, J.M. and Dryden, D.T.F. (2002). Structure of Ocr from Bacteriophage T7, a Protein that Mimics B-Form DNA, *Mol. Cell* **9**, pp. 187-194.
- Woese, C.R., Olsen, G.J., Ibba, M. and Söll, D. (2000). Aminoacyl-tRNA Synthetases, the Genetic Code and the Evolutionary Process, *Microbiol. Mol. Biol. Rev.* **64**, pp. 202-236.
- Wu, H.-L., Bagby, S. and van den Elsen, J.M.H. (2005). Evolution of the Genetic Triplet Code via Two Types of Doublet Codons, *J. Mol. Evol.* **61**, pp. 54-64.
- Yaremcuk, A., Krikliyyi, I., Tukhalo, M. and Kusack, S. (2002). Class I Tyrosyl-tRNA Synthetase has a Class II mode of tRNA Recognition, *EMBO J.* **21**, pp. 3829-3840.
- Yarus, M., Caporaso, J.G. and Knight, R. (2005). Origins of the Genetic Code: The Escaped Triplet Theory, *Ann. Rev. Biochem.* **74**, pp. 179-198.

# Index

- amino acids, 10
  - classes, 13, 19
  - doubling, 24
  - R-groups, 10
- aminoacyl-tRNA synthetases, 12
- biochemical assembly, 15, 24
- biochemical information, 2
- decoherence, 17, 26
- doublet code, 20
- enzymes, 27
- evolution, 4, 29
  - cooperation, 6
  - direction, 6
  - mutation, 5
  - selection, 5
- frozen accident, 5
- genetic code, 17
  - duplication, 18
  - evolution, 22
  - mitochondrial, 20
  - predecessor, 21
  - universal, 20
- genetic information, 4
- genetic languages, 7
- Grover's algorithm, 14, 16, 24
- information theory, 2, 28
- knowledge, 3, 7
- language, 3
  - efficiency, 3
- life, 2
- minimal language, 8
  - DNA, 14
  - proteins, 9
- operational RNA code, 19, 22
- optimal solution, 5
- polypeptide chains, 10
- protein folding, 12, 13
- Ramachandran map, 10
- ribozyme, 18, 22
- RNA world, 18, 29
- structural language, 11
- superposition, 16, 25
- tetrahedral geometry, 9
- transition states, 27
- tRNA, 12, 18
  - acceptor stem, 22
- wave mechanics, 15
- wobble pairing, 20
- wobble position, 22, 23